

A NEW CLASSIFICATION ALGORITHM: OPTIMALLY GENERALIZED LEARNING VECTOR QUANTIZATION (OGLVQ)

*T. Temel**

Abstract: We present a new Generalized Learning Vector Quantization classifier called Optimally Generalized Learning Vector Quantization based on a novel weight-update rule for learning labeled samples. The algorithm attains stable prototype/weight vector dynamics in terms of estimated current and previous weights and their updates. Resulting weight update term is then related to the proximity measure used by Generalized Learning Vector Quantization classifiers. New algorithm and some major counterparts are tested and compared for synthetic and publicly available datasets. For both the datasets studied, it is seen that the new classifier outperforms its counterparts in training and testing with accuracy above 80% its counterparts and in robustness against model parameter variation.

Key words: *machine learning, classification, learning vector quantization, self-organized mapping, supervised learning, unsupervised learning*

Received: May 11, 2015

DOI: 10.14311/NNW.2017.27.031

Revised and accepted: January 4, 2018

1. Introduction

A successful classifier algorithm is expected to be able to generalize varying realization of a given data distribution with high-accuracy even if data is of noisy contamination and its learning/training stage to be stable and of low-complexity. High-accuracy objective is mostly achieved by directing the learning process in favor of correct class (reward) and against the incorrect class (punishment) in order to minimize misclassification even if data is rarely available or prone to false judgment [1]. Conventional Learning Vector Quantization (LVQ) and its variants have been considered a group of the most successful classifiers, which are well-suited to above objectives [9, 10]. They are based on Hebbian-learning with winner-take-all and the nearest-neighborhood attributed to by class-associated prototype/weight vectors in feature space. Despite simplicity in weight update rule in training, the main problem with standard LVQ classifiers stems from representative proximity or (dis)similarity measure in learning or training and its appropriate practice in classification. Therefore, it is mandatory to formulate an adaptation scheme as a

*Turgay Temel; Bursa Technical University, Faculty of Natural Sciences, Architecture and Engineering, Mechatronics Engineering Dept. Osmangazi, Bursa – Turkey, E-mail: turgay.temel@btu.edu.tr, ttemel170@gmail.com

precaution against unstable behavior in weight update since prototype vector for a loser class, which might have been incorrectly assigned during training are forced to diverge from respective samples. Another problem with standard LVQ classifiers is initialization of weight vectors. Major remedial solutions have focused on developing an adaptive metric for expressing dissimilarities during training. For example, Generalized Learning Vector Quantization (GLVQ) realizes learning subject to a hypothesis-testing based cost function being optimized with use of gradient descent as a metric to represent overall decision-making performance [11]. Although generalization capability of GLVQ is almost independent of data representation for small dimensions, weight vectors exhibit saliency as data dimension increases and stability in weight update becomes a hindering factor. Also, proximity measure used with GLVQ will fail in processing highly correlated data and overlapped classes since the metric exploited may fail in representing sample-to-weight proximities [4, 7]. A possible remedy to this shortcoming, which may be attributed to a functional relevance, is to utilize kernel-based scaling for proximities of individual samples to weight vectors, which is coined Generalized Relevance Learning Vector Quantization (GRLVQ) [3, 7]. Training phase of GRLVQ also aims at optimizing a decision-making cost function similar to GLVQ training. Although major (G/GR)LVQ algorithms consider ensemble of prototype-data proximities as a measure of learning performance, to our knowledge, there has been no algorithm up-to-date which provides conditions on stability, and consistency for weight/prototype weight vectors and their update given arbitrary initialization and robustness to parametric variation.

In this paper, we present a new GLVQ classifier called Optimally Generalized Learning Vector Quantization (OGVLQ) based on a new training scheme associated to a convergence rule for stable weight dynamics and consistent learning. Proposed rule, then, utilized the proximity measure exploited by previous (G/GR)LVQ algorithms. New and some major (G)LVQ algorithms are tested and compared for synthetic and publicly available datasets for statistical performance measures. Results show that the new LVQ is faster in training and assigns correct classes more successfully with improved robustness against variation in learning parameter and randomness in initialization on weight vectors compared to the predecessors it is compared for the datasets studied.

2. LVQ Classification: An Overview

A generic operation of an LVQ classifier, which is shown in Fig. 1, is to assign class labels which are known a priori to input samples to be classified. For a sample L -dimensional input column vector $\mathbf{x} = [x_1 \dots x_L]^T$ with superscript ‘T’ referring to transposition and given K class labels $C_{i=1,\dots,K}$, where the weight vector of the j -th class label C_j is $\mathbf{w}^j = [w^{j1} \dots w^{jL}]^T$, the output class label which corresponds to a winner is assigned. The winning output vector \mathbf{w}^* is determined as the closest weight vector, i.e. $\mathbf{w}^* = \arg d(\mathbf{x}_k, \mathbf{w}^j)$, where $d(\mathbf{x}_k, \mathbf{w}^j)$ is the squared Euclidean distance between \mathbf{x}_k and \mathbf{w}_j .

In training phase of original LVQ [8], with N_{train} samples per epoch, for the k -th input sample vector $\mathbf{x}_k = \mathbf{x}(k)$, $k = 1, \dots, N_{\text{train}}$, only the weight vector of

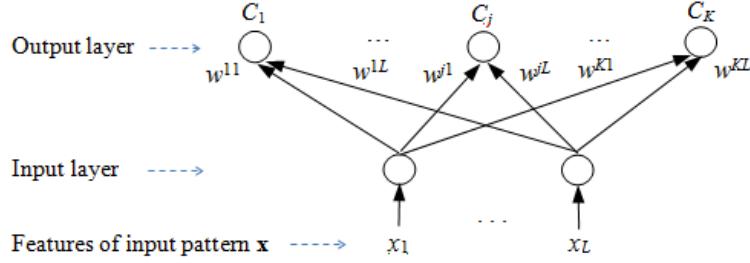


Fig. 1 Generalized operation of LVQ classifiers.

the winning class label, $\mathbf{w}_k^* = \mathbf{w}^*(k)$, is adjusted by

$$\mathbf{w}_{k+1}^* = \mathbf{w}_k^* \pm \xi (\mathbf{x}_k - \mathbf{w}_k^*) \quad (1)$$

until a pre-specified convergence condition is met, e.g. $\mathbf{w}^j(k) = \mathbf{w}^j(k-1)$ for $j = 1, \dots, K$. The parameter ξ refers to learning rate that may be updated in iteration, e.g. decreasing with the number of iterations/epochs of training. The sign ‘±’ is taken ‘+’ if \mathbf{x}_k has been correctly classified otherwise ‘−’ such that the winning weight vector is driven toward the data if class label is correctly identified or vice versa, respectively. Other variants of original LVQ, e.g. LVQ2.1 and LVQ3, adjust weight vectors of some other classes, e.g. closest loser prototype of having the same class label as the sample [13, 14].

On the other hand, training of (G/GR)LVQ classifiers involves optimizing a cost function which relates correctly classified samples to particular class weight vectors to establish a proximity measure to maximize correct decision as a supervisory operation. Such a cost function is given by

$$E = \frac{1}{2} \sum_{\forall k} f(\mu_k), \quad (2)$$

where the classifier function is $f(u) = \frac{1}{1+e^{-u}}$ for a measure of proximity $\mu_k = \mu(\mathbf{x}_k) = \frac{d_k^+ - d_k^-}{d_k^+ + d_k^-}$. Dissimilarity measures $d_k^+ = d(\mathbf{x}_k, \mathbf{w}_k^* = \mathbf{w}^+) = \|\mathbf{x}_k - \mathbf{w}^+\|^2$ and $d_k^- = d(\mathbf{x}_k, \mathbf{w}_k^* = \mathbf{w}^-) = \|\mathbf{x}_k - \mathbf{w}^-\|^2$ are the squared distances of \mathbf{x}_k to the closest prototype \mathbf{w}^+ with the same class label as \mathbf{x}_k and the best matching prototype \mathbf{w}^- with a class label different from that of \mathbf{x}_k , respectively. The weight update is then implemented as

$$\mathbf{w}_{k+1}^* \leftarrow \mathbf{w}_k^* \pm \underbrace{\xi(\mu_k)(\mathbf{x}_k - \mathbf{w}_k^*)}_{\Delta \mathbf{w}_k^*}. \quad (3)$$

It is seen that (G/GR)LVQ algorithms adopt varying $\xi(\mu_k) = \frac{\partial f}{\partial \mu_k} \frac{d_k^\pm}{(d_k^+ + d_k^-)^2}$ to project spatial localization of weights with respect to sample \mathbf{x}_k for improved accuracy [6].

3. New GLVQ Algorithm

When training samples have not been pruned with overlapped classes and/or initial weights are chosen improperly, weights and updates may not converge, which may lead to instability in learning [2]. Thus, it is desirable to provide a method for ensuring stable learning through weight update dynamics. For this objective, a reasonable approach is to minimize the cost function

$$J = \left(\frac{1}{2} \right) \sum_k \|\Delta \mathbf{w}_k^*\|^2, \quad (4)$$

which is satisfied with decreased gradient of J . To obtain a monotonic decrease in gradient J it is required to maintain $\|\Delta \mathbf{w}_k^*\|^2 < \|\Delta \mathbf{w}_{k-1}^*\|^2$. We propose a relationship $\|\Delta \mathbf{w}_k^*\|^2 = \|\Delta \mathbf{w}_{k-1}^*\|^2 \pm \alpha_k (\|\mathbf{w}_k^*\|^2 - \|\mathbf{w}_{k-1}^*\|^2)$ given sample and a respective decision-dependent parameter $|\alpha_k| < 1$, where $\Delta \mathbf{w}_{k-1}^* = \mathbf{w}_k^* - \mathbf{w}_{k-1}^*$ and $\Delta \mathbf{w}_k^* = \mathbf{w}_{k+1}^* - \mathbf{w}_k^*$. The sign ‘±’ is taken ‘-’ if \mathbf{w}_k^* and \mathbf{w}_{k-1}^* both refer to the same class label otherwise ‘+’. Thus, it is possible to relate \mathbf{w}_k^* (\mathbf{w}_{k-1}^*) and $\Delta \mathbf{w}_k^*(\Delta \mathbf{w}_{k-1}^*)$ for stable dynamics. We consider the convergence rule

$$(\Delta \mathbf{w}_k^*)^T \mathbf{w}_k^* \leq -\eta \|\mathbf{w}_k^*\|^2 \quad (5)$$

similar to the approach in [5, 15, 16], where $\eta > 0$. By using $\mathbf{w}_{k-1}^* = \mathbf{w}_k^* - \Delta \mathbf{w}_{k-1}^*$ (5) can be rewritten as

$$(\Delta \mathbf{w}_k^*)^T \mathbf{w}_k^* \mp \eta \left\{ \frac{\|\Delta \mathbf{w}_k^*\|^2 - \|\Delta \mathbf{w}_{k-1}^*\|^2}{\alpha_k} \right\} \pm (\Delta \mathbf{w}_{k-1}^*)^T \mathbf{w}_{k-1}^*, \quad (6)$$

where the operator sequence ‘ $\leq -, +$ ’ ($>, +, -$) is considered from left to right if \mathbf{w}_k^* and \mathbf{w}_{k-1}^* refer to the same (different) class label(s). Equating the right-hand side of (6) to 0 as a boundary for convergence and combining the resulting expression with that of (5) corresponding to $(k-1)$ -th term will yield

$$\|\Delta \mathbf{w}_k^*\|^2 = \|\Delta \mathbf{w}_{k-1}^*\|^2 (1 \mp \alpha_k). \quad (7)$$

For convenience, it is appropriate to consider $\alpha_k = \mu_k$ for relating new learning algorithm to spatial saliency in decision making exploited by (G/GR)LVQ classifiers. Then, use of gradient for (7) leads to

$$(\Delta \mathbf{w}_k^*)^T = (\Delta \mathbf{w}_{k-1}^*)^T \left(1 \mp \alpha_k \mp \frac{\Delta \mathbf{w}_{k-1}^*}{(d_k^+ + d_k^-)^2} \left[d_k^+ \frac{\partial d_k^+}{\partial \mathbf{w}_k^+} - d_k^- \frac{\partial d_k^-}{\partial \mathbf{w}_k^-} \right] \right), \quad (8)$$

where $\frac{\partial d_k^\pm}{\partial \mathbf{w}_k^\mp} = 2(\mathbf{x}_k - \mathbf{w}^\pm)^T$ with sign previously given for (G/GR)LVQ. The signs ‘ $-, -$ ’ ($+, +$) refers to \mathbf{w}_k^* and \mathbf{w}_{k-1}^* be of same (different) class label(s) in guiding the weight update and class labeling rules along with convergence rule during training.

4. Experiments

New classifier, and its previously cited original LVQ, LVQ2.1, and GLVQ counterparts were designed and compared for accuracy in testing and training time complexity with two sets of experiments for $\xi = 0.02$, $\xi = 0.05$, and $\xi = 0.1$, respectively. The chosen values of ξ are to simulate the training and convergence rate characteristics of designed classifiers against parameter variations, which may be considered as resulting from a suitable adaptive model [12].

The first set of experiments serves solely for visualizing the capability of the new algorithm in predicting the class labels of test samples given overlapping synthetic density profiles with use of random subsampling. For this purpose, 100 experiments were performed for each ξ value. At each experiment, 100 random vectors $\mathbf{x} = [x_1 \ x_2]$ were populated from each of five 2D (bivariate) normal densities where 20 (80) samples were set aside per class in forming a training (testing) dataset, i.e. the total number of training (testing) samples $N_{\text{train}} = 100$ ($N_{\text{test}} = 400$). The class labels and respective densities are $C_1: N([1.5 \ 0], [1 \ -0.5; -0.5 \ 1])$, $C_2: N([-1.5 \ 0], [1 \ 0.75; 0.75 \ 1])$, $C_3: N([1.5 \ -1.5], [1 \ 0.5; 0.5 \ 1])$, $C_4: N([0 \ -1.5], [1 \ -0.75; -0.75 \ 1])$, and $C_5: N([0 \ 1.5], [1 \ 0.5; 0.5 \ 1])$ where the first quantity in parentheses is the mean vector while the second is the covariance matrix with rows separated by semicolon. As an illustrative example, scatter for a testing sample dataset and respective classification results with original and proposed LVQ classifiers for $\xi = 0.05$ are shown in Fig. 2 Some important statistical measures, such as average and standard deviation in accuracy, i.e. classification success score in testing, and training time complexity in terms of number of epochs, are given in Tab. I.

Algorithm	Accuracy or classification success in testing, %,			# of training epochs, Average/std. deviation		
	Average/std. deviation					
	$\xi = 0.02$	$\xi = 0.05$	$\xi = 0.1$	$\xi = 0.02$	$\xi = 0.05$	$\xi = 0.1$
OGLVQ	83.8 / 3.1	84.5 / 2.7	81.1 / 3.0	43.1 / 4.6	36.3 / 3.2	35.2 / 3.4
LVQ	53.5 / 4.8	46.1 / 5.1	51.5 / 5.0	61.4 / 5.9	59.2 / 6.2	51.1 / 5.6
LVQ2.1	72.6 / 4.1	63.5 / 4.5	67.4 / 4.4	57.3 / 6.2	50.5 / 5.7	47.5 / 5.3
GLVQ	74.7 / 3.2	77.8 / 3.9	71.8 / 4.2	47.9 / 4.6	40.6 / 5.2	43.7 / 5.2

Tab. I Statistical performance measures of new (OGLVQ) and some other LVQ algorithms for 2D normal densities chosen.

From above figure, it is seen that despite considerable overlap between classes, new algorithm is capable of identifying even those test samples that fall in interclass regions. On the other hand, original LVQ algorithm creates Voronoi-like regions, i.e. performing clustering operation rather than classification, which means that it is unable to classify such samples and issues incorrect class labels for those samples.

The second set of experiments employs cross-validation with average of $K = 10$ subgroup outputs as K -fold model building and evaluating the performance of respective classifiers studied. For this purpose, Character Trajectories Dataset, which is publicly available at [17], was used. The dataset consists of 3-dimensional 2858

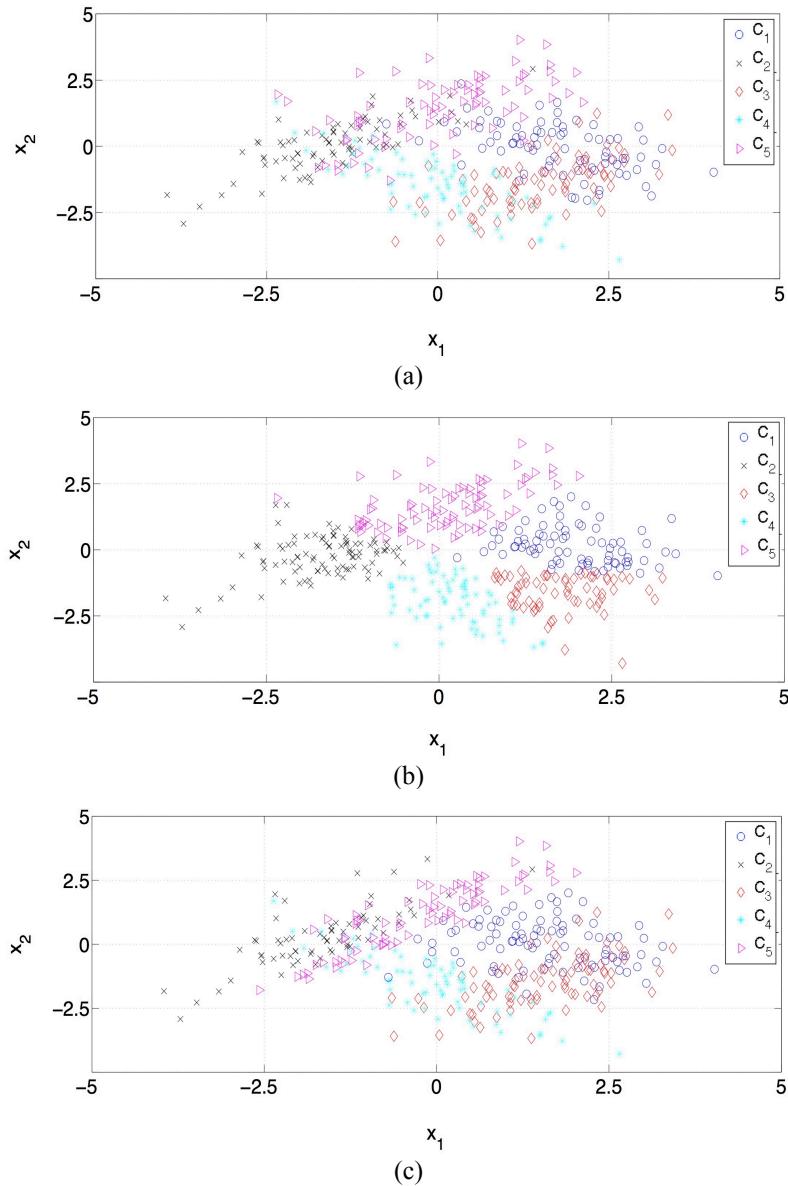


Fig. 2 Example scatter of (a) a testing dataset of 2D normal densities, and classification results with (b) original LVQ, and (c) proposed algorithms for $\xi = 0.05$.

labeled samples of pen tip segment trajectories for the 20 single pen-down characters, e.g. ‘a’, ‘e’, ‘w’. The feature vectors are composed of respective coordinates x , y , and pen tip force. From the dataset, randomly chosen 2850 samples from the dataset were utilized Cross-validation was implemented by setting one out of ten subgroups of the dataset for testing while the remaining nine was used for model

building or training. Tab. II presents some major characteristics of the classifiers studied in statistical terms, i.e. average and statistical deviation of classification accuracy in testing and training time complexity.

Algorithm	Accuracy or classification success in testing, %,			# of training epochs, Average/std. deviation		
	Average/std. deviation			$\xi = 0.02$	$\xi = 0.05$	$\xi = 0.1$
	$\xi = 0.02$	$\xi = 0.05$	$\xi = 0.1$	$\xi = 0.02$	$\xi = 0.05$	$\xi = 0.1$
OGLVQ	85.6 / 3.3	82.8 / 3.3	84.6 / 3.0	42.3 / 3.6	28.6 / 2.5	31.8 / 3.0
LVQ	51.9 / 5.0	48.6 / 4.5	56.9 / 4.3	56.2 / 4.7	46.9 / 4.3	42.1 / 4.0
LVQ2.1	70.1 / 4.4	70.9 / 4.8	66.6 / 4.7	47.7 / 5.0	40.8 / 4.6	39.6 / 4.4
GLVQ	78.9 / 4.1	79.2 / 3.7	81.0 / 3.8	48.9 / 4.2	44.3 / 4.2	42.5 / 4.0

Tab. II Statistical performance measures of new (OGLVQ) and some other LVQ algorithms for Character Trajectories Dataset with for $K = 10$ fold cross-validation.

From Tabs. I and II, it is seen that for both datasets, the OGLVQ classifier exhibits better and consistent statistical characteristics in training and testing classification performance against variation in learning parameter and random initialization in weight vectors than the other LVQ classifiers for the datasets utilized. Although GLVQ classifier exhibits comparable performance in classifying test samples only for $\xi = 0.05$, OGLVQ outperforms its counterparts it is compared to in statistical terms of training, generalization and robustness against model parameter variation.

5. Conclusions

We presented a new Generalized Learning Vector Quantization classifier called Optimally Generalized Learning Vector Quantization. New algorithm is based on a weight-update rule for training. The algorithm seeks stability in prototype/weight vector dynamics and their updates. Resulting weight update term is then associated to the proximity measure used by Generalized Learning Vector Quantization classifiers. New algorithm and some major counterparts were tested and compared for synthetic and publicly available datasets. Results indicate that new classifier is faster in training and is more successful and robust in classifying test samples of datasets studied than the counterparts it is compared.

References

- [1] BACKHAUS A., SEIFFERT U. Classification in high-dimensional spectral data: Accuracy vs. Spectral data: Accuracy vs. interpretability vs. model size. *Neurocomputing*. 2014, 131(5), pp. 15–22, doi: [10.1016/j.neucom.2013.09.048](https://doi.org/10.1016/j.neucom.2013.09.048).
- [2] BOUBEZOUL A., PARIS S., OULADSINE M. Application of the cross entropy method to the GLVQ algorithm. *Pattern Recognition*. 2008, 41, pp. 3173–3178, doi: [10.1016/j.patcog.2008.03.016](https://doi.org/10.1016/j.patcog.2008.03.016).

- [3] CATARON A., ANDONIE R. Energy Generalized LVQ with Relevance Factors. In: *Proceedings of IEEE International Joint Conference on Neural Networks (IJCNN 2004)*. 2004, pp. 1421–1426, doi: [10.1109/IJCNN.2004.1380159](https://doi.org/10.1109/IJCNN.2004.1380159).
- [4] HAMMER B., STRICKERT M., VILLMANN T. On the generalization ability of GRLVQ networks. *Neural Processing Lett.* 2005, 21(2), pp. 109–120, doi: [10.1007/s11063-004-1547-1](https://doi.org/10.1007/s11063-004-1547-1).
- [5] JANARDHANAN S., BANDYOPADHYAY B. On Discretization of Continuous-Time Terminal Sliding Mode. *IEEE Trans. Automatic Control*. 2006, 51(9), pp. 1532–1536, doi: [10.1109/TAC.2006.880805](https://doi.org/10.1109/TAC.2006.880805).
- [6] KADEN M., LANGE M., NEBEL D., RIEDEL M., GEWENIGER T., VILLMANN T. Aspects in Classification Learning-Review of Recent Developments in Learning Vector Quantization. *Foundations of Computing and Decision Sciences*, 2014, 39(1), pp. 79–105, doi: [10.2478/fcds-2014-0006](https://doi.org/10.2478/fcds-2014-0006).
- [7] KÄSTNER M., HAMMER B., BIEHL M., VILLMANN T. Functional relevance learning in generalized learning vector quantization. *Neurocomputing*. 2012, 90, pp. 85–95, doi: [10.1016/j.neucom.2011.11.029](https://doi.org/10.1016/j.neucom.2011.11.029).
- [8] KOHONEN T. Self-Organized Formation of Topologically Correct Feature Maps. *Biological Cybernetics*. 1982, 43(1), pp. 59–69, doi: [10.1007/bf00337288](https://doi.org/10.1007/bf00337288).
- [9] LLOYD G.R., BRERETON R.G., FARIA R., DUNCAN J.C. Learning Vector Quantization for Multiclass Classification: Application to Characterization of Plastics. *Jour. of Chem. Info. Model.* 2007, 47(4), pp. 1553–1563, doi: [10.1021/ci700019](https://doi.org/10.1021/ci700019).
- [10] NOVA D., ESTÉVEZ P. A review of learning vector quantization classifiers. *Neural Computing and Applications*, 2013, 25(3-4), pp. 511–524, doi: [10.1007/s00521-013-1535-3](https://doi.org/10.1007/s00521-013-1535-3).
- [11] SATO A., YAMADA K. Generalized learning vector quantization. *Advances in Neural Information Processing Systems*. 1996, pp. 423–429. Available from: <https://papers.nips.cc/paper/1113-generalized-learning-vector-quantization.pdf>.
- [12] SCHMIDT M., BABANEZHAD R., AHMED M.O., DEFAZIO A., CLIFTON A., SARKAR A. Non-Uniform Stochastic Average Gradient Method for Training Conditional Random Fields. *arXiv preprint*. 2015. Available from: [arXiv:1504.04406](https://arxiv.org/abs/1504.04406) [stats.ML].
- [13] TEMEL T., KARLIK B. An improved odor recognition system using learning vector quantization with a new discriminant analysis. *Neural Network World*. 2007, 17(4), pp. 287–294.
- [14] TEMEL T. Biologically-inspired Learning: An Overview and Application to Odor Recognition. In: T. TEMEL, ed. *System and Circuit Design for Biologically-Inspired Intelligent Learning*. Hershey, PA, USA: IGI Global, 2010, pp. 59–92, doi: [10.4018/978-1-60960-018-1.ch004](https://doi.org/10.4018/978-1-60960-018-1.ch004).
- [15] TEMEL T., ASHRAFIUON H. Sliding-mode control approach for faster tracking. *IET Elect. Lett.* 2012, 48(15), pp. 916–917, doi: [10.1049/el.2012.1576](https://doi.org/10.1049/el.2012.1576).
- [16] TEMEL T., ASHRAFIUON H. Sliding-mode speed controller for tracking of underactuated surface vessels with extended Kalman filter. *IET Elect. Lett.* 2015, 51(6), pp. 467–469, doi: [10.1049/el.2014.4516](https://doi.org/10.1049/el.2014.4516).
- [17] WILLIAMS B.L. UCI Machine Learning Repository, Irvine, CA, *Character Trajectories Data Set*. 2008, School of Informatics, University of Edinburgh, UK. Available from: <http://archive.ics.uci.edu/ml/machine-learning-databases/character-trajectories/>.